# Машинное обучение.

Алексей Колесов

Белорусский государственный университет

4 сентября 2017 г.

# Содержание

# Устройство курса

- 14 лекций (темы и можно увидеть на странице курса `http://anytask.org/course/207`)
- 8 семинарских занятий
- зачёт

# Домашние задания

14 домашних заданий — по одному после каждой лекции

- работа сдана в течение 7 дней — **2** балла
- работа сдана в течение 14 дней — **1** балл
- работа не сдана в течение 14 дней — **0** баллов

Кроме того, в некоторых домашних работах будут встречаться задания повышенной сложности, которые могут повысить вашу оценку (при условии, что домашняя работа сдана за две недели).

# Семинарские занятия

На семинарских занятиях мы будем решить некоторые задачи по пройденному материалу (необязательно по теме лекции этого дня).
Типичный семинар будет проходить следующим образом:

- я выдаю набор задач
- вы решаете эти задачи (можно в группах)
- за каждую решенную задачу решившим ставится $\frac{k}{n}$ балла, где $n$ — количество решивших, $k$ — стоимость задачи (округление вниз до десятых)
- за каждую нерешённую задачу в семинаре **всем** (у кого нет официальной причины для отсутствия) **вычитается 1 балл**

# Зачёт

**Допуск к зачёту**

- зачесть как минимум 11 домашних заданий
- набрать как минимум 17 баллов

Каждая несданная домашка означает задание по теме домашней работе на зачёте.

**Зачёт автоматом**

- решить все домашки
- набрать 21 балл

# Дополнительные замечания

- посещение лекций и семинаров не влияет на оценивание
- компьютер вам понадобится лишь для решения домашних заданий
- лекционный материал зачастую сложней семинарских и домашних заданий
- в этом курсе очень строгое отношение к дедлайнам!

# Содержание

# Что такое машинное обучение

**Том Митчелл**

A computer program is said to learn from experience **E**, with respect to some task **T**, and some performance measure **P**, if its performance on **T** as measured by **P** improves with experience **E**.



Что такое обучение - одного определения нет.

Задумываться стали. 1950-е, Самуэль Артур, шашки, лучше чем он сам

Идея любого определения - преобразование опыт в знание

# Примеры обучения в окружающем мире

- крысы *учатся* избегать отравленные приманки (если было плохо после маленького кусочка, больше такое не есть)
- эксперимент с голубями[1] показывает, что они суеверны (готовы делать что угодно, если это приносит еду)
- фундаментальный вопрос: как нам различить полезное обучение от суеверия
- эксперименты с крысами[2] показывают, что они не видят связи между вкусом еды и ударами тока; или между звуком и плохим самочувствием

[1] http://psychclassics.yorku.ca/Skinner/Pigeon
[2] Garcia & Koelling 1996



Крысы сначала откусывают лишь небольшой кусочек приманки, чтоб проверить, как еда на них влияет. Будут ли они дальше есть зависит от вкуса еды и эффекта на самочувствие.

Мы можем так же построить систему определения спама - запомнить плохие сообщения и не пропускать их во входящие. Плохо - нет обобщения на невиденные раньше письма. Крысы обобщают - если еда похожа по вкусу или запаху - такое не едят. Мы тоже можем - например, найти слова, которые часто встречаются в спаме.

Ученый Скиннер провёл эксперимент - клетка с голубями - даётся еда вне зависимости от поведения. Кто-то кивал, кто-то стоял на одной ноге, кто-что. Голуби начинали повторять это действие чаще, тем самым вероятность, что следующий раз он будет делать при еде то же самое растёт. В итоге, они начинали делать это постоянно.

Как различить обучение и суеверие. Вернёмся к крысам. Эксперименты двух товарищей говорят, что если заменить еду на звук или плохое самочувствие на удар током - то крысы не находят связи. А значит у них есть априорное знание.

# Некоторые понятия

**Обобщающая способность** — качество программы показывать хорошее качество на примерах, которые она не видела раньше

**Inductive bias** — набор предположений (априорных знаний), который используется для предсказания неизвестных значений

Для успешного обучения использование априорных знаний неизбежно (No Free Lunch theorem).

# Зачем нужно машинное обучение

- задачи, которые сложно запрограммировать
  - сложноформализуемые задачи (например, распознавание символов, речи, вождение автомобиля)
  - задачи неподвластные человеку (анализ астрономических данных, ранжирование веб-страниц)
- задачи, для которых нужна адаптация

# Типы обучения

- с учителем и без учителя
- активное и пассивное
- онлайн и оффлайн

С учителем и без - например определение спама (у нас есть метки) против детектирования аномалий на ядерной станции (у нас нет меток и их дорого произвести)
Влияем ли мы на исходные данные или нет
Дан ли нам сразу все данные, либо даются по одному
Мы почти весь курс про пассивное оффлайн обучение с учителем.
Вообще говоря, с учителем и без учителя можно делить дальше.
Есть semisupervised - когда даны и размеченные данные и не размеченные. И есть ещё вариант transduction - когда на трейне даны с метками, а тест просто дан (но без меток). Например, интернет страницы.
Вопрос: чем может быть полезно? Тем, что у нас есть больше информации про распределение теста.

# Типовые задачи машинного обучения

- классификация
- регрессия
- ранжирование
- кластеризация
- уменьшение размерности

# О чём этот курс

- математические основы машинного обучения
- обзор алгоритмов машинного обучения
- анализ алгоритмов с точки зрения их производительности и гарантий

Мы будем отвечать на вопросы: что можно выучить, при каких условиях, какие гарантии на качество

# Что в этом курсе не будет

- обзор библиотек машинного обучения
- советов, как победить на Kaggle
- нейронных сетей (разве что на свободной лекции)

# Содержание

# Формальная модель обучения

- $X$ — множество объектов (**domain set**); характеризуется набором признаков (**features**)
- $Y$ — множество меток (**label set**); например, вкусный/не вкусный $\{0, 1\}$
- $S = ((x_1, y_1), \ldots, (x_m, y_m))$, $x_i \in X$, $y_i \in Y$ — тренировочные данные, тренировочная выборка (**training set**); $S|_x = (x_1, \ldots, x_m)$ — набор объектов в тренировочной выборке
- $h : X \to Y$ — ответ алгоритма, гипотеза, решающее правило (**hypothesis, predictor**)
- $A$ — алгоритм машинного обучения: $A(S) = h$



Для того, чтоб ответить на поставленные вопросы нужно ввести некую модель обучения. Для начала мы сделаем ряд упрощающих предпложений, а затем будем их убирать, тем самым получая более полезную модель.
Давайте представим, что мы приехали в какой-нибудь неизвестный нам ранее город Беларуси и узнали, что там в основном едят драники нового для нас сорта. Как настоящие беларусы мы тоже хотим покушать драники, но перед покупкой всё-таки хочется научится определять, какие драники будут вкусными, а какие не очень.
Для начала нужно определить по каким качествам (или признакам) мы будем определять вкусноту. По нашему большому опыту поедания драников мы можем предпложить, что вкус зависит от двух параметров — размера драника и его цвета.
Что мы можем использовать для того, чтоб научиться предсказывать вкус драников? Для этого можно купить какой-то набор драников разных параметров и попробовать. Таким образом у нас будут некоторые данные, а именно — последовательность драников с пометкой для каждого - вкусный он был или нет.

# Простая модель генерации данных

- Предположим, что существует распределение вероятности $D$ над множеством объектов $X$
- Каждый объект тренировочной выборки $x_i \in S|_x$ выбирается независимо из $D$ (предположение о независимости и одинаковой распределённости)
- Существует функция $f : X \to Y$, $y_i = f(x_i)\ \forall i$ (предположение о детерминированности среды)
- Ошибка классификации: $L_{D,f}(h) = \mathop{\mathbb{P}}_{x \sim D}[h(x) \neq f(x)]$ (**generalization error**, **true risk**)

Тут следует объяснить, откуда именно взялась тренировочная выборка.

Предположение о независимости и одинаковой распределённости иногда тоже можно убрать (domain adaptation)

Про детерминированность среды уберём в течение этой лекции

# Информация, доступная алгоритму

- алгоритм знает только про выборку $S$
- у него нет никаких знаний про $D$ (тут отличие от математической статистики)
- у него нет никаких знаний про $f$ (более того, ровно её мы и хотим найти)

# Минимизация эмпирического риска

**Модель**: алгоритм принимает $S$, полученный из распределения $D$ и размеченный функцией $f$. Его задача найти гипотезу $h_S : X \to Y$, который минимизирует ошибку $L_{D,f}(h_S)$ по отношению к **неизвестным** $D$ и $f$.

- $D$ и $f$ неизвестны $\Rightarrow L_{D,f}(h_s)$
- давайте использовать ошибку на тренировочной выборке (**empirical risk, empirical error**):

$$L_S(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

**Минимизация эмпирического риска** — парадигма обучения, заключающаяся в выборе гипотезы, минимизирующей ошибку на тренировочной выборке

# Переобучение

На практике, минимизация эмпирического риска зачастую приводит к переобучению (**overfitting**).

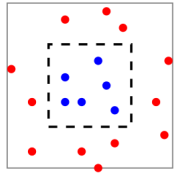

Рис.: Пример распределения драников

Пример (*вероятно, плохой*) ERM-гипотезы:

$$h_S(x) = \begin{cases} y_i & \text{если } \exists i \in [m] \text{ такой что } x_i = x \\ 0 & \text{иначе} \end{cases}$$



Несмотря на очень естественную идею минимизации эмпирического риска, на практике это работает не так уж хорошо.
Фундаментальная проблема заключается в том, что конечная выборка не может содержать в себе все детали потенциально очень сложного распределения $D$. В итоге, алгоритм может найти гипотезу, которая хорошо объясняет тренировочную выборку, но плохо объясняет множество объектов
Давайте вернёмся к примеру с драниками. Пусть по отношению к нашим двум параметрам все драники равномерно распределены в таком квадрате. Причем множество вкусных (обозначенных синим) - расположено в центральном квадрате, площадью вдвое меньше, чем весь domain set.
Рассмотрим пример ERM-гипотезы. Она говорит, что если встретили драник ровно такой, что был в выборке, то ответим тот же ответ, что у нас есть. Иначе, скажем, что драник невкусный.
Конечно, такая гипотеза выглядит немного странной, но мы увидим, что она вполне может возникнуть, как полиномиальная формула от фичей.
Так вот, очевидно, что на тренировочной выборке эта гипотеза не ошибается. Но на всём $D$ у неё будет loss равный единице.
Интуитивно, переобучение возникает, когда мы слишком хорошо ведём себя на тренировочной выборке.

# ERM with inductive bias

- ERM-правило приводит к переобучению
- Вместо того, чтоб не использовать его, найдём случаи, когда это правило работает достаточно хорошо
- Хороший способ — ограничить набор гипотез
- $H$ — семейство гипотез из $X$ в $Y$;

$$\mathrm{ERM}_H(S) \in \operatorname*{argmin}_{h \in H} L_S(h)$$

- Один из важных вопросов машинного обучения: «для каких $H$ $\mathrm{ERM}_H$ не переобучается»



Достаточно хорошо - то есть, если хорошо работает на тренировочных данных, то с высокой вероятностью хорошо работает и на всех данных

Будем поступать как крысы в упомянутом эксперименте - будем рассматривать не все возможные гипотезы, а лишь какое-то подмножество

Ограничение семейства гипотез происходит ещё до обучения, а значит должно содержать какие-то априорные знания о задаче. Например, мы можем предположить, что все вкусные драники находятся в каком-то подпрямоугольнике пространства размер-цвет.

Мы чуть позже докажем, что в таком классе $\mathrm{ERM}_H$ не переобучается.

С другой стороны, если $H$ — все возможные гипотезы, то переобучение неизбежно

Надо упомянуть про tradeoff: ошибка против сложности класса

# ERM для конечного набора гипотез

- Пусть набор гипотез $H$ — конечен
- Сделаем следующее предпложение

**Предположение реализуемости** — найдётся такая гипотеза $h^* \in H$, что $L_{D,f}(h^*) = 0$

- Предположение значит, что (с вероятностью 1) найдётся гипотеза $h \in H$, что $L_S(h) = 0$
- Но нам интересен $L_D(h)$, а не $L_S(h)$
- Предположим, что $S \sim D^m$ (предположение о независимости и одинаковой распределённости)



Ограничение в конечность набора гипотез может рассматриваться как не очень жёсткое: например, мы можем сказать, что это лишь гипотезы, которые могут бысть выражены не более чем 10kb C++-ным файлом. Набор всех прямоугольников - тоже бесконечный класс, но мы можем сказать, что они задаются четырьмя числами в памяти компьютера, каждое число - 64 бита. Тогда всего у нас есть $2^256$ гипотез.

Предположение реализуемости очень сильное. Однако мы сможем избавиться от него чуть позже.

# ERM для конечного набора гипотез

- $S$ — случайная величина $\Rightarrow h_S = ERM_H(S)$ — случайная величина $\Rightarrow L_{D,f}(h_S)$ — случайная величина
- Нереалистично предполагать, что мы всегда сможем получить хороший $L_{D,f}(h_S)$ (нам может очень не повезти с выборкой)
- Кроме того, даже если с выборкой повезло, её конечность всё равно означает то, что полностью $D$ мы оценить не сможем

Например, нам могут попасть только невкусные драники.

# ERM для конечного набора гипотез

Чтоб оценить эти замечания, введём два параметра:

- $\delta$ — вероятность того, что нам попадётся плохая (нерепрезентативная) выборка (число $1 - \delta$ называется **confidence parameter**)
- $\epsilon$ — максимальная ошибка, которую мы разрешаем иметь нашему классификатору на $D$ (**accuracy**)
- $L_{D,f}(h_S) > e$ — неуспех алгоритма

# ERM для конечного набора гипотез

Мы хотим оценить сверху следующую вероятность:

$$D^m(\{S|_x : L_{D,f}(h_S) > e\})$$

Введём набор «плохих» гипотез:

$$H_B = \{h \in H : L_{D,f}(h) > e\}$$

И набор «опасных» тренировочных выборок:

$$M = \{S|_x : \exists h \in H_B, L_S(h) = 0\}$$

Из-за предположения о реализуемости:

$$\{S|_x : L_{D,f}(h_S) > e\} \subseteq M$$

Опасная выборка - такая, что на ней есть гипотеза, которая хороша на выборке, но плоха в общем

Т.е. множество, вероятность которого мы хотим оценить, лежит внутри опасных выборок

# ERM для конечного набора гипотез

Мы имеем:

$$M = \{S|_x : \exists h \in H_B, L_S(h) = 0\}$$

$$\{S|_x : L_{D,f}(h_S) > e\} \subseteq M$$

Перепишем $M$:

$$M = \bigcup_{h \in H_B} \{S|_x : L_S(h) = 0\}$$

Отсюда:

$$D^m(\{S|_x : L_{D,f}(h_S) > e\}) \leqslant D^m \left( \bigcup_{h \in H_B} \{S|_x : L_S(h) = 0\} \right)$$

# ERM для конечного набора гипотез

**Union bound**

Для любых двух событий и любого распределения вероятностей:

$$\mathbb{P}(A \cup B) \leqslant \mathbb{P}(A) + \mathbb{P}(B)$$

$$D^m\left(\bigcup_{h \in H_B} \{S|_x : L_S(h) = 0\}\right) \leqslant \sum_{h \in H_B} D^m\left(S|_x : L_S(h) = 0\right)$$

$$D^m(\{S|_x : L_{D,f}(h_S) > e\}) \leqslant \sum_{h \in H_B} D^m\left(S|_x : L_S(h) = 0\right)$$

# ERM для конечного набора гипотез

Оценим каждое слагаемое в предыдущей сумме

$$D^m(S|_x : L_S(h) = 0) = D^m(S|_x : \forall i, h(x_i) = f(x_i)) = \\ = \prod_{i=1}^{m} D(\{x : h(x) = f(x)\})$$

Так как $h \in H_B$:

$$D(\{x : h(x) = f(x)\} = 1 - L_{D,f}(h) \leqslant 1 - \epsilon$$

Так как $1 - \epsilon \leqslant e^{-\epsilon}$:

$$D^m(S|_x : L_S(h) = 0) \leqslant (1 - \epsilon)^m \leqslant e^{-\epsilon m}$$

# ERM для конечного набора гипотез

Соберём вместе:

$$D^m(\{S|_x : L_{D,f}(h_S) > e\}) \leqslant \sum_{h \in H_B} D^m(S|_x : L_S(h) = 0)$$
$$\leqslant |H_B| e^{-\epsilon m} \leqslant |H| e^{-\epsilon m}$$

# ERM для конечного набора гипотез

Пусть $H$ — конечный набор гипотез. Возьмём произвольные $\delta \in (0, 1)$ и $\epsilon > 0$ и число $m$, такое что:

$$m \geqslant \frac{\log(|H|/\delta)}{\epsilon}$$

Тогда для любой функции $f$, для любого распределения $D$ (с выполненным предположением о реализуемости) с вероятностью как минимум $1 - \delta$ ERM-гипотеза $h_S$ от выборки $S$ размера $m$, порождённой независимо распределением $D$ и размеченной функцией $f$, выполняется:

$$L_{D,f}(h_S) \leqslant \epsilon$$

Таким образом, наша гипотеза вероятно (с вероятностью $1 - \delta$) приблизительно (с ошибкой не больше $\epsilon$) верна

# Probably approximately correct learnability

Класс гипотез $H$ называют **вероятно приблизительно верно изучаемым** (probably approximately correct learnable) если существует такая функция $m_H : (0,1)^2 \to \mathbb{N}$ и алгоритм, такой что

- для любых $\epsilon, \delta \in (0,1)$
- для любого распределения $D$ над $X$
- для любой функции $f : X \to \{0,1\}$

если выполняется предположение о реализуемости, то если мы выполним алгоритм на выборке из $m \geqslant m_H(\epsilon, \delta)$ независимых одинаково распределённых элементов из $D$ и размеченных $f$, то алгоритм вернёт гипотезу $h \in H$ такую, что с вероятностью как минимум $1 - \delta$, выполняется $L_{D,f}(h) \leqslant \epsilon$



В определении два параметра приближения - от них не избавиться.

## Выборочная сложность

Функция $m_H$ называется выборочной сложностью (**sample complexity**)
Зависит от:

- $\epsilon$ — accuracy
- $\delta$ — confidence
- $H$ (например, для конечных классов мы видели, что зависит от $\log |H|$)

Любой конечный класс является PAC-изучаемым с выборочной сложностью

$$m_H(\epsilon, \delta) \leqslant \left\lceil \frac{\log\left(|H|/\delta\right)}{\epsilon} \right\rceil$$

Для любого класса таких функций много. Мы зачастую хотим «минимальную»
Бывают и бесконечные PAC-learnable классы

# Обобщение PAC-learnable model

- Избавимся от предположения о реализуемости
- Добавим в модель не только бинарную классификацию

# Избавление от предположения о реализуемости

- Пусть теперь $D$ — распределение над $X \times Y$, например $D(x, y) = D_x(x) \cdot D((x, y)|x)$
- True risk:

$$L_D(h) = \mathop{\mathbb{P}}_{(x,y)\sim D}[h(x) \neq y] = D(\{(x, y) : h(x) \neq y\})$$

- Эмпирический риск:

$$L_S(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

Глупо предполагать, что найдётся гипотеза в нашем классе, которая всегда имеет правильный ответ. Может быть сам мир стохастичен, а может быть нам не хватает данных: например, вкус драника определятеся чем-то кроме размера и цвета. Придётся обобщить модель, чтоб учесть наши замечания
Можно говорить, что $D$ — совместное распределение
Посмотрим, как меняются показатели качества нашего классификатора при такой модели порождения данных

# Оптимальный байесовский классификатор

Наша цель — найти гипотезу, у которой $L_D(h)$ будет (вероятно приблизительно) низким.

**Оптимальный байесовский классификатор**

Классификатор:

$$f_D(x) = \operatorname*{argmax}_y D(x, y)$$

называется **оптимальным байесовским классификатором**

Данный классификатор называется оптимальным, в том смысле, что он доставляет минимальную ошибку функционалу true risk. Т.е. любой другой классификатор показывает ошибку не меньшую, чем данный.

Беда в том, что мы не можем использовать этот классификатор, ведь у нас нет информации о $D$.

В общем, у нас нет надежды найти что-то лучшее, чем байесовский классификатор. Более того, если у нас нет никаких априорных знаний о распределении, у нас даже нет надежды найти что-то с таким же лоссом, как он.

Поэтому, мы можем лишь расчитывать на то, что наша ошибка будет небольшой по сравнению с теми, что есть у нас в классе гипотез

# Agnostic PAC-learnable

Класс гипотез $H$ называют **агностически вероятно приблизительно верно изучаемым** (agnostic PAC-learnable) если существует такая функция $m_H : (0,1)^2 \to \mathbb{N}$ и алгоритм, такой что

- для любых $\epsilon, \delta \in (0,1)$
- для любого распределения $D$ над $X \times Y$
- ~~для любой функции $f : X \to \{0,1\}$~~

~~если выполняется предположение о реализуемости, то~~ если мы выполним алгоритм на выборке из $m \geqslant m_H(\epsilon, \delta)$ независимых одинаково распределённых элементов из $D$ ~~и размеченных $f$~~, то алгоритм вернёт гипотезу $h \in H$ такую, что с вероятностью как минимум $1 - \delta$, выполняется $L_{D,f}(h) \leqslant \min\limits_{h' \in H} L_D(h') + \epsilon$

Если раньше мы хотели хорошую ошибку в абсолютном значении, то теперь лишь относительно лучшей гипотезы из класса, так как мы не можем гарантировать достаточно хорошей ошибки в любом случае. Тем не менее, если предположение реализуемости выполняется, то Agnostic PAC-learnable то же самое, что просто PAC-learnable – поэтому обобщение.

# Обобщённые функции потерь

- Пусть дан набор гипотез $H$ и некий домен $Z$. Пусть есть функция $l : H \times Z \to \mathbb{R}_+$; такую функцию будем называть *функцией потерь* (**loss function**)
- True risk: $L_D(h) = \mathop{\mathbb{E}}_{z \sim D}[l(h, z)]$
- Empirical risk: $L_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h, z_i)$
- **0-1 loss** $l_{01}(h, (x, y)) = [h(x) = y]$
- **Square loss** $l_{sq}(h, (x, y)) = (h(x) - y)^2$

Для задач предсказания (классификации и регрессии) $Z = X \times Y$.

# Agnostic PAC-leanable for generalized loss functions

Класс гипотез $H$ называют **агностически вероятно приблизительно верно изучаемым** (agnostic PAC-learnable) по отношению к множеству $Z$ и функции потерь $l : H \times Z \to \mathbb{R}_+$, если существует такая функция $m_H : (0,1)^2 \to \mathbb{N}$ и алгоритм, такой что

- для любых $\epsilon, \delta \in (0,1)$
- для любого распределения $D$ над $Z$

если мы выполним алгоритм на выборке из $m \geqslant m_H(\epsilon, \delta)$ независимых одинаково распределённых элементов из $D$, то алгоритм вернёт гипотезу $h \in H$ такую, что с вероятностью как минимум $1 - \delta$, выполняется $L_{D,f}(h) \leqslant \min\limits_{h' \in H} L_D(h') + \epsilon$, где $L_D(h) = \mathop{\mathbb{E}}\limits_{z \sim D}[l(h, z)]$



Стоит заметить, что мы хотим получать гипотезу из класса $H$. Однако, есть методы машинного обучения, когда мы гипотезу получаем из большого класса $H'$, но гарантия на точность остаётся про $H$. Такое называется representation independent learning (а обычный сетап - proper learning). Хотя нельзя сказать, что representation independent чем-то improper

# Содержание

# Идея обучения через равномерную сходимость

- ERM-алгоритм выбирает самую лучшую гипотезу по отношению к тренировочной выборке
- True risk для такой гипотезы может оказаться гораздо выше, чем empirical risk
- **Идея**: давайте найдём такие условия, что у любой гипотезы из класса true risk несильно отличается от empirical risk

Тренировочная выборка $S$ называется $\epsilon$-**репрезентативной** (по отношению к домену $Z$, классу гипотез $H$, функции потерь $l$ и распределению $D$), если

$$\forall h \in H, |L_D(h) - L_S(h)| \leqslant \epsilon$$



Вопрос к аудитории. Где используется $Z$? В $l$, которая используется в $L_D$. Ну и в самой $D$.

# Лемма о ERM в случае $\frac{\epsilon}{2}$-репрезентативной выборки

**Лемма о ERM в случае $\frac{\epsilon}{2}$-репрезентативной выборки**

Пусть выборка $S$ является $\frac{\epsilon}{2}$-репрезентативной. Тогда если $h_S$ — ERM-гипотеза, то выполняется:

$$L_D(h_S) \leqslant \min_{h \in H} L_D(h) + \epsilon$$

Доказательство будет на семинаре.

Если вдруг найдутся условия, что выборка является $\epsilon/2$-репрезентативной с вероятностью $1 - \delta$, то класс гипотез будет PAC-learnable.

## Равномерная сходимость

Класс гипотез $H$ обладает свойством **равномерной сходимости** (**uniform convergence**) (по отношению к домену $Z$ и функции потерь $l$), если существует такая функция $m_H^{UC} : (0,1)^2 \to \mathbb{N}$, что для любых $\delta$, $\epsilon$ из $(0,1)$ и любого распределения $D$ над $Z$, если выборка $S$ состоит из $m \geqslant m_H^{UC}(\epsilon, \delta)$ объектов, выбранных из $D$ независимо, то с вероятностью как минимум $1 - \delta$, $S$ является $\epsilon$-репрезентативной выборкой.

Если класс $H$ обладает свойством равномерной сходимости с выборочной сложностью $m_H^{UC}$, то этот же класс является агностически PAC-изучаемым с выборочной сложностью $m_H(\epsilon, \delta) \leqslant m_H^{UC}(\epsilon/2, \delta)$. Более того, в этом случае ERM-алгоритм — успешный PAC-learner

Где в сходимости вообще используется $H$?

# Доказательство PAC-learnability для конечных классов

Хотим показать, что:

$$D^m(\{S : \forall h \in H, |L_S(h) - L_D(h)| \leqslant \epsilon\}) \geqslant 1 - \delta$$

План следующий:

1. Применим union bound
2. Оценим каждый элемент суммы

# Доказательство PAC-learnability для конечных классов: шаг 1

Хотим:

$$D^m(\{S : \forall h \in H, |L_S(h) - L_D(h)| \leqslant \epsilon\}) \geqslant 1 - \delta$$

То же самое, что:

$$D^m(\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon\}) < \delta$$

Опять перепишем:

$$\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon\} = \bigcup_{h \in H} \{S : |L_S(h) - L_D(h)| > \epsilon\}$$

# Доказательство PAC-learnability для конечных классов: шаг 1

Хотим:

$$D^m(\{S : \forall h \in H, |L_S(h) - L_D(h)| \leqslant \epsilon\}) \geqslant 1 - \delta$$

Получили:

$$D^m(\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon\}) \leqslant \sum_{h \in H} D^m(\{S : |L_S(h) - L_D(h)| > \epsilon\})$$

# Доказательство PAC-learnability для конечных классов: шаг 2

- Надо показать, что для любой выбранной $h$ величина $|L_D(h) - L_S(h)|$ с высокой вероятностью небольшая относительно выбора $S$
- Напомним, что $L_D(h) = \underset{z \sim D}{\mathbb{E}}[l(h, z)]$, а $L_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h, z_i)$
- $\mathbb{E}[L_S(h)] = L_D(h)$ — очень полезное наблюдение!
- А значит, $|L_D(h) - L_S(h)|$ — отклонение случайной величины $L_S(h)$

# Доказательство PAC-learnability для конечных классов: шаг 2

- Нам нужно оценить, насколько величина сконцентрирована относительного своего среднего
- Закон больших чисел говорит, что выборочное среднее стремится к своему среднему, но это лишь асимптотический закон

**Неравенство Хёфдинга**

Пусть $\theta_1, \ldots, \theta_m$ — последовательность независимых одинаково распределённых на $[a; b]$ случайных величин с $\mathbb{E}[\theta_i] = \mu$. Тогда для любого $\epsilon > 0$:

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right| > \epsilon\right] \leqslant 2\exp(-2m\epsilon^2/(b-a)^2)$$



что будем использовать в качестве $\theta$ ?

# Доказательство PAC-learnability для конечных классов: шаг 2

Пусть $\theta_i = l(h, z_i)$; тогда $L_S(h) = \frac{1}{m}\sum_{i=1}^{m}\theta_i$, $L_D(h) = \mu$

Также, предположим, что $l$ имеет значения только в $[0; 1]$.

$$D^m\left(\{S : |L_S(h) - L_D(h)| > \epsilon\}\right) = \\ = \mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right| > \epsilon\right] \leqslant \\ \leqslant 2\exp(-2m\epsilon^2)$$

# Доказательство PAC-learnability для конечных классов

А значит:

$$D^m(\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon\}) \leqslant 2|H| \exp(-2m\epsilon^2)$$

Таким образом, если взять

$$m \geqslant \frac{log(2|H|/\delta)}{2\epsilon^2}$$

То:

$$D^m(\{S : \exists h \in H, |L_S(h) - L_D(h)| > \epsilon\}) \leqslant \delta$$

# Содержание

# Итоги

- Сегодня мы сделали обзор основных понятий машинного обучения
- Построили основную модель обучения — agnostic PAC-learnability
- Разработали инструмент для доказательства agnostic PAC-learnability — uniform convergence
- Доказали, что ERM-алгоритм является agnostic PAC-learnable для конечного класса гипотез